John T. Anderson

Engineering Note

Date: June 14,2000
Rev Date: ~~June 14,2000~~ **July 10,2000**

Project: D-Zero Central Fiber Tracker
Doc. No: A1000614

Subject: Internal calibration of A/D converters on AFE, AFE_Test, MCM_Test and Meltronix boards

Introduction

The internal microcontroller used on the Analog Front End (AFE) board, its test module and the two MCM testing boards (MCM_TEST and Meltronix) is the PIC14000 from Microchip. This microcontroller uses an external capacitor driven by a programmable current source to implement a Wilkinson, or slope, A/D converter. In the AFE module this A/D converter is used to monitor board temperature, VLPC bias voltage, VLPC current draw and cassette temperature. Due to variation between capacitors and temperature/aging drift, online calibration of the A/D converter will be periodically required. This engineering note addresses the conversion technique used within the board and details a procedure by which the converter may be calibrated. In this analysis the number and type of parameters which must be shared between a given AFE board and the remote monitoring system for purposes of calibration will be derived.

Change History

July 10, 2000: typographical errors in Table 2 repaired.

Conversion Methodology – Software Perspective

The microcontroller on the AFE responds to eight-bit numeric command codes which are written to the board from the MIL-STD 1553 interface. Various commands result in A/D conversions which are presented back to the MIL-STD 1553 interface as sixteen-bit unsigned quantities, where the numeric value corresponds approximately to a voltage read by the A/D from zero to five volts. Various prescaling amplifiers set external signals to fit this nominal range. To perform an A/D conversion one simply writes the appropriate command to the microcontroller and the desired information is written to specific addresses accessible from the MIL-STD 1553 interface.

Calibration Issues

The A/D conversion read by the external software is not ideal. Various gain errors and offset errors are possible, mostly due to component variation. Figure 1 gives a sketch of the generic problem, pointing out where error will creep in.

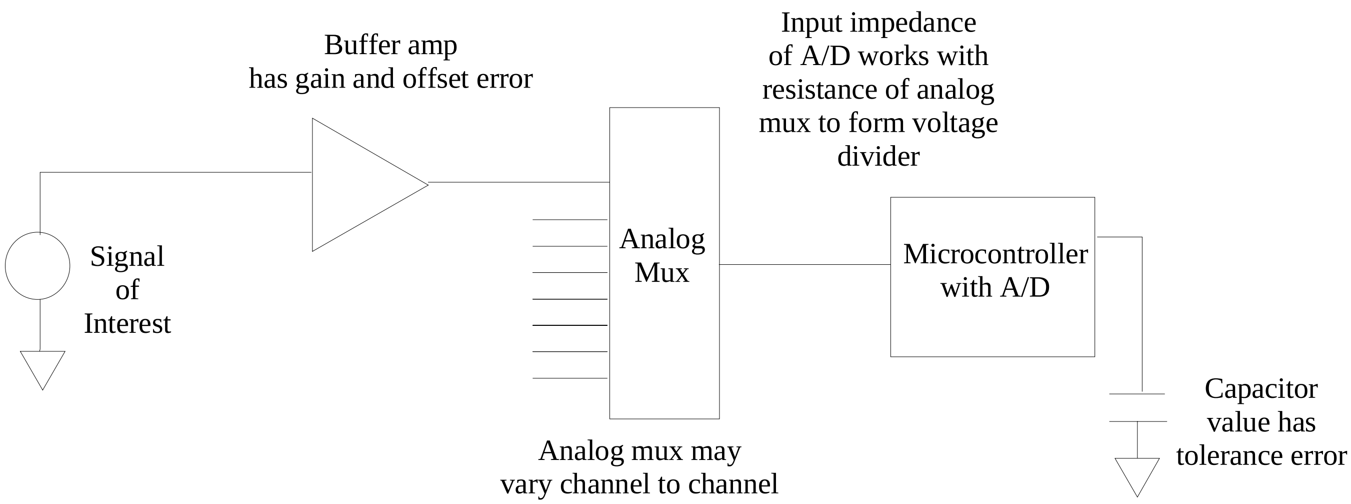

Figure 1

The tolerance error of the capacitor can be compensated for, at least to first order, by the microcontroller. The hardware conversion process utilizes a constant current source to charge the capacitor during the conversion process, and the current used may be selected from a set of values. This allows a coarse adjustment of the gain of the A/D converter such that all AFE boards have roughly the same conversion gain. However, it's not conceivable that the limited number of currents available will allow all the gain error from the capacitor to be completely zeroed out. Thus, a method of calculating the *actual* gain of the A/D process is required.

The PIC14000 provides us with a number of internal reference voltages which may be used to select the best current source value and to measure the exact conversion gain of the A/D itself. Figure 2, taken from the data sheet of the PIC14000, shows the internal structure of the A/D converter, especially the internal analog mux which provides access to different internal references.

FIGURE 8-1: A/D BLOCK DIAGRAM

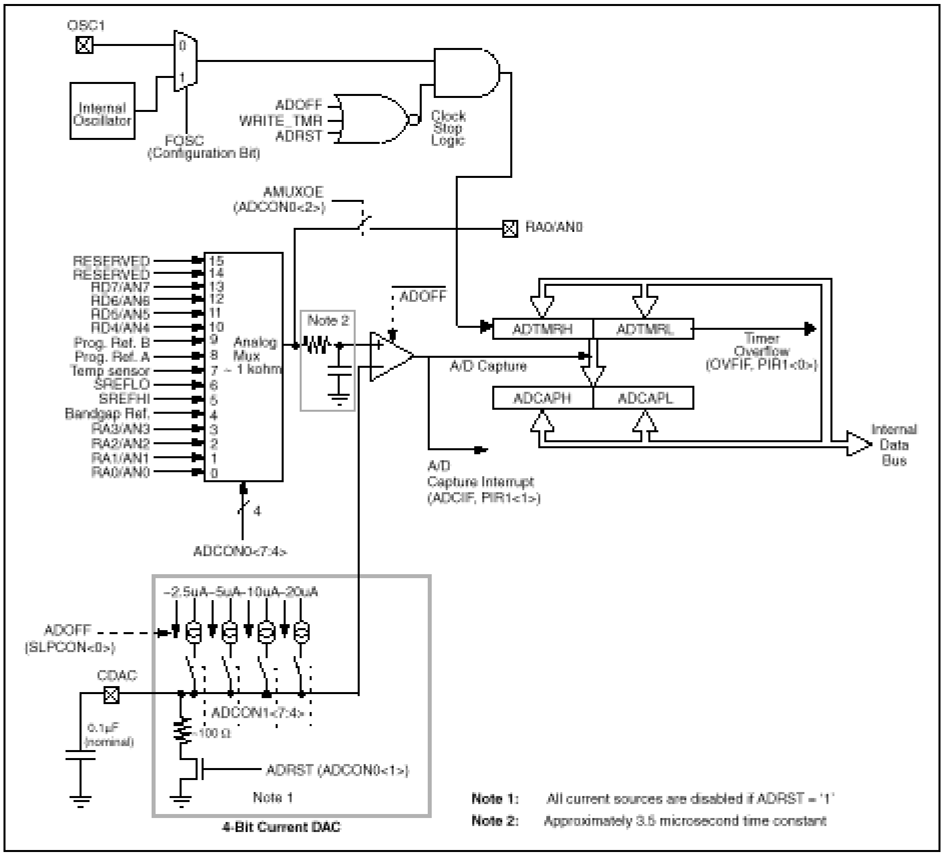

Figure 2

The Bandgap Reference input provides a nominal 1.2V test voltage. Two Slope References are provided internally which give a divided output of the same Bandgap Reference, such that SREFHI is nominally the same as the Bandgap and SREFLO is nominally 0.13V. By measuring these voltages one should be able to determine the slope of the A/D converter gain in counts per volt, and adjust the current source appropriately to achieve something close to 65,536 counts per 5 volts. Verification of the accuracy of this fit can be achieved, *but only on the test bench, not in place,* by two means:

1. One pin of the microcontroller which can be used as an analog input is brought out to a two pin header on the board. An accurate power supply or DAC may be connected to this pin and the exact transfer function known. The conversion constants so obtained may then be burned into the microcontroller's EPROM. When the board is in service in the field, the SREFLO/SREFHI readouts may then be compared to the 'test bench' standard to determine if drift has occurred.
2. In similar fashion it is possible to route any analog signal presented to the PIC14000 back out to an unused I/O pin of the device, providing a point by which a DVM may be used to obtain calibration.

Internal Firmware Routines Necessary

A hierarchical set of routines are required in the PIC14000 to provide the necessary functionality. All routines may be linked to the 1553 bus via the command processing routine for engineering diagnostics, although it is expected that the final release may not allow experiment monitoring system such uncontrolled access.

DO_CONVERSION

This routine would simply do one ADC conversion, assuming that the internal and external analog multiplexers were previously set up by some other code. Additionally, this routine assumes that the current source is also correctly set up prior to execution. The routine would simply discharge the ramp capacitor, let it ramp up, read the ADC conversion value and return that value in two internal PIC14000 'f' registers (as the ADC count is a sixteen bit value).

CALIBRATE_GAIN

This routine would perform calibration setup by using the bandgap reference. The routine would do the following sequence:

1. Select the bandgap as the source of the analog mux.
2. Set the current source to the minimum value.
3. Call DO_CONVERSION.
4. Compare the result of the conversion to the expected value (a constant), based upon the known voltage of the bandgap reference and a desired ADC gain of 5.00 Volts = maximum ADC count.
5. If the result of the conversion is too high, then select the next higher current source value and loop back to step 3. If conversion value is too low or within tolerance, exit to step 6.
6. If result was within tolerance, store current source selector value into dual port ram. Also store ADC conversion result in dual-port ram. Store a value in dual-port ram indicating successful calibration.
7. If result was too low for minimum current source value, store error code into dual port ram, but also store current source selector value and ADC conversion result to dual-port ram.
8. Select the SREFLO as the source of the analog mux. Loop back to step 2 and repeat process, using different comparison constant, and storing results into different block of dual port ram addresses.
9. Select the PIC14000 power supply as the source of the analog mux.
10. Loop back to step 2 and repeat process, using only one current source setting, the average of the two settings determined in the last two passes. Save the current source setting used for this conversion and the result of the conversion into the dual port ram.

This process should leave a block of data in the dual port ram for access by the experiment monitoring system that looks like Table 1. When the CALIBRATE_GAIN routine is called, it fills the entire block. The three ‘expected value’ routines are, of course, simply taken from constants in the firmware and reported back to make the job of the experiment monitor system easier.

An error location is found at the end of the list which tells the experiment monitoring system if any problems or inconsistencies were encountered in the calibration process. See table 1 for details.

# of bytes	Value stored	Use by microcontroller
2	Setting of constant current source as 16-bit value (high byte always zero) determined for conversion of 1.3V bandgap source	Filled with data by routine
2	Actual conversion value returned from conversion of bandgap source using current setting in previous word	Filled with data by routine
2	Expected conversion value for conversion of 1.3V bandgap source (if everything was ideal)	Filled with data by routine
2	Setting of constant current source as 16-bit value (high byte always zero) determined for conversion of SREFLO bandgap source (nominal 0.13V)	Filled with data by routine
2	Actual conversion value returned from conversion of SREFLO bandgap source using current setting in previous word	Filled with data by routine
2	Expected conversion value for conversion of SREFLO (nominal 0.13V) bandgap source (if everything was ideal)	Filled with data by routine
2	Setting of constant current source as 16-bit value (high byte always zero) determined for conversion of +5V source	Filled with data by routine
2	Actual conversion value returned from conversion of +5V supply using current setting in previous word	Filled with data by routine
2	Expected conversion value for conversion of +5V supply bandgap source (if everything was ideal)	Filled with data by routine
2	Error flag location. Set to zero (0x0000) if everything works. Each nibble (4-bit group) of this location is set to 0xF to denote that an error occurred. If multiple bit groups are set, multiple errors have occurred. xxxF indicates an error in the conversion of the 1.3V bandgap reference xxFx indicates an error in the conversion of the SREFLO bandgap reference xFxx indicates that the selected 'best' current source value for 1.3V reference and SREFLO reference conversions was not the same value. Fxxx indicates overflow error in conversion of +5V supply when converted.	Filled with data by routine

Table 1

SET_CURRENT_SOURCE

This routine simply accesses a byte at a predetermined location in the dual port RAM and puts that value in the current source control register. This default current source value is used for all experiment monitor conversions.

CONVERT_SELECTED_DEVICE

This routine gets an address byte from a predetermined location in the dual port RAM, interprets it as a device address, then calls DO_CONVERSION to perform an A/D on that device. The resulting 16-bit value is then placed in a predetermined location in dual-port memory. This addressing scheme is not trivial. The address byte must first select whether the object to be converted is internal to the PIC14000 or external. If external, then this routine must utilize the bus selection logic implemented in the Helper CPLD and the I^{2}C bus serial protocol implemented in other PIC14000 firmware routines.to connect the appropriate input to the A/D converter prior to initiating the conversion.

The bitmap for the address byte goes as follows:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Unused (0)	External analog mux A2	External analog mux A1	External analog mux A0	Internal analog mux A3	Internal analog mux A2	Internal analog mux A1	Internal analog mux A0

The bottom four bits select which of the 16 possible inputs to the PIC14000 internal analog mux is utilized. Some of the possible 16 codes connect to PIC internals such as the bandgap, whereas others map through the pins to different devices. When the code selects an external device, the external device will be an eight-channel analog mux, requiring selection of one of the eight inputs of that external mux from bits 4 through 6. Implicit in this method is that the value of bits 3:0 will give a bus address for the external analog mux, selecting which of the three eight-channel analog muxes on the AFE is selected. Table 2 sorts it all out.

Value from address byte	Device selected	Other implcit addressing required
Binary 0xxx0000	Pin AN0 of PIC	None. This pin is not connected to any analog input.
Binary 0nnn0001	Pin AN1 of PIC	Pin AN1 is connected to the VLPC Bias Readback analog mux. The I2C selection register must be set to connect the I2C pins of the PIC to bus SDA3/SCL3. The I2C device select byte must be set to I2C device address 10010100. The I2C channel select byte must be driven from bits 6:4 of the address byte (the nnn in the value at left).
Binary 0nnn0010	Pin AN2 of PIC	Pin AN2 is connected to the VLPC Current Readback analog mux. The I2C selection register must be set to connect the I2C pins of the PIC to bus SDA3/SCL3. The I2C device select byte must be set to I2C device address 10010000. The I2C channel select byte must be driven from bits 6:4 of the address byte (the nnn in the value at left).
Binary 0nnn0011	Pin AN3 of PIC	Pin AN3 is connected to the Cryostat Temperature Readback analog mux. The I2C selection register must be set to connect the I2C pins of the PIC to bus SDA3/SCL3. The I2C device select byte must be set to I2C device address 10010010. The I2C channel select byte must be driven from bits 6:4 of the address byte (the nnn in the value at left).
Binary 0xxx0100	PIC Bandgap Reference	None.
Binary 0xxx0101	PIC SREFHI Reference	None.
Binary 0xxx0110	PIC SREFLO Reference	None.
Binary 0xxx0111	PIC Temperature Sensor	None.
Binary 0xxx1000	PIC Programmable Ref. A	None.
Binary 0xxx1001	PIC Programmable Ref. B	None.
Binary 0xxx1010	Pin RD4/AN4 of PIC.	*Do not use. This pin is reserved for digital usage.*
Binary 0xxx1011	Pin RD5/AN5 of PIC.	*Do not use. This pin is reserved for digital usage.*
Binary 0xxx1100	Pin RD6/AN6 of PIC.	*Do not use. This pin is reserved for digital usage.*
Binary 0xxx1101	Pin RD7/AN7 of PIC.	*This pin may be used for engineering tests only, only on the test bench, and requires certain modifications to the Helper CPLD program load. Do not use under normal conditions.*
Binary 0xxx1110	Reserved in PIC	*Do not use under any circumstances.*
Binary 0xxx1111	Reserved in PIC	*Do not use under any circumstances.*

Table 2

READ_ALL_VLPC_VOLTAGES

This routine makes a set of calls to CONVERT_SELECTED_DEVICE to obtain readbacks of all eight VLPC Bias Readback values, which are stored in the dual port ram as a contiguous.block of eight 2-byte values.

READ_ALL_VLPC_CURRENTS

This routine makes a set of calls to CONVERT_SELECTED_DEVICE to obtain readbacks of all eight VLPC Current Readback values, which are stored in the dual port ram as a contiguous.block of eight 2-byte values.

READ_ALL_CRYO_TEMPS

This routine makes a set of calls to CONVERT_SELECTED_DEVICE to obtain readbacks of all eight Cryostate Temperature Readback values, which are stored in the dual port ram as a contiguous.block of eight 2-byte values.